

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A networked computer comprising an unsolicited message diverting communications processor connected between mail transfer agents MTA_0 with an Internet address IP_0, a from-address A_0, a declared domain D_0, and a real domain DD_0, and MTA_1 with an Internet address IP_1, a domain D_1, a to-address A_1, a diversion address A'_1, and a save_spam database, the unsolicited message diverting communications processor comprising:

- a) monitoring means for monitoring the communications between MTA_0 and MTA_1;
- b) determining means for determining if the communications contains an unsolicited message; and
- c) intercepting means for intercepting a RCPT reply from MTA_0, substituting the diversion address A'_1 for the to-address A_1 in the RCPT reply and sending a modified RCPT reply to MTA_1 if the message is determined to be unsolicited and if the to-address A_1 is in the save_spam database;

wherein the unsolicited message diverting communications processor does not intercept communications between MTA_0 and MTA_1 before a RCPT command from MTA_0 is received by the unsolicited message diverting communications processor, and

wherein the connection with MTA_0 is rejected before the data portion of the unsolicited message is transmitted.

2. (Currently amended) The networked computer ~~unsolicited message diverting~~ communications processor in Claim 1, further includes an allow_address database and wherein the determining means determines if a message is not unsolicited by checking if the IP_0 is in the allow_address database.

3. (Currently amended) The networked computer ~~unsolicited message diverting~~ communications processor in Claim 1, further includes a prevent_address database and wherein

the determining means determines if a message is unsolicited by checking if IP_0 is in the prevent_address database.

4. (Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, further includes access to an open relay database and wherein the determining means determines if a message is unsolicited by checking if IP_0 is in the open relay database.
5. (Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, further includes access to a DNS (domain name server) database and wherein the determining means determines if a message is unsolicited by checking if IP_0 has a domain name entry DD_0 in the DNS database.
6. ((Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, further includes a bad_from database and wherein the determining means determines if a message is unsolicited by checking if the from-address A_0 is in the bad_from database.
7. (Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, further includes a suspect_domain database and wherein the determining means determines if a message is unsolicited by checking if the real domain DD_0 matches the domain of the from-address A_0 and the domain of the from-address A_0 is in the suspect_domain database.
8. (Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the from-address A_0 matches the to-address A_1.
9. (Currently amended) The networked computer ~~unsolicited-message-diverting communications-processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D_0 of MTA_0 is the same as the domain D_1 of MTA_1.

10. (Currently amended) The networked computer ~~unsolicited-message-diverting communications processor~~ in Claim 1, wherein the determining means determines if a message is unsolicited by checking if the declared domain D_0 of MTA_0 does not match the real domain DD_1DD_0 and the declared domain D_0 is in the suspect_domain database.

11. (Currently amended) The networked computer ~~unsolicited-message-diverting communications processor~~ in Claim 1, further includes a no_filter database and wherein the determining means determines if an unsolicited message should be blocked by checking if the to-address A_1 is in the no_filter database.

12. (Currently amended) The networked computer ~~unsolicited-message-diverting communications processor~~ in Claim 1, further includes a rejected_connection database which logs the time, from-address A_0, to-address A_1, and the reason for the rejection if a message is rejected if the message is determined to be unsolicited.

13. (Currently amended) The networked computer ~~unsolicited-message-diverting communications processor~~ in Claim 1, further includes an allowed_connection database which logs the time and to-address A_1 if the message is determined not to be unsolicited.

14. (Currently amended) A method implemented in a networked computer for a receiving networked computer system with an Internet connection, a mail transfer agent MTA_1, an Internet address IP_1, a to-address A_1, a diversion address A'_1, a save_spam database and an operating system capable of executing the method to divert unsolicited messages from a transmitting networked computer system with an Internet connection and a mail transfer agent MTA_0, an Internet address IP_0, a from-address A_0, a declared domain D_0, and a real domain DD_0 comprising the steps of:

- a) waiting for a new SMTP connection request;
- b) relaying and monitoring the replies from MTA_0 to MTA_1;
- c) relaying replies from MTA_1 to MTA_0;
- d) intercepting a RCPT reply from MTA_0 to MTA_1;
- e) determining if the message is unsolicited by analyzing the monitored replies;
- f) releasing the intercepted RCPT reply if the message is determined not to be

unsolicited; and

g) substituting the diversion address A'_1 for the to-address A_1 in the RCPT reply and sending the modified reply to MTA_1 if the message is determined to be unsolicited and if the to-address A_1 is in the save_spam database;

whereby MTA_1 controls the interaction between MTA_0 and MTA_1 before a RCPT command from MTA_0 is received and whereby the connection with MTA_0 is rejected before the data portion of the unsolicited message is transmitted.

15. (Currently amended) A method implemented in a networked computer system for a receiving networked computer system with an Internet connection, a mail transfer agent MTA_1, an IP address IP_1, a domain name D_1, a to-address, A_1, a recipient diversion address A'_1, an allow_address database, a prevent_address database, a suspect_domain database, a bad_from database, a no_filter database, a rejected_connection database, an allowed_connection database, a save_spam database, a diversion database, and an operating system capable of executing the method to divert unsolicited messages from a transmitting networked computer system with an Internet connection, a mail transfer agent MTA_0, an IP address of IP_0, a declared domain name D_0, a real domain name DD_0, and a from-address of A_0 comprising the steps of:

a) waiting for a SMTP connection request on the receiving networked computer system's Internet connection;

b) sending a 220 reply to MTA_0 to acknowledge the requested SMTP connection request;

c) extracting IP_0 from the SMTP connection request;

d) requesting the real domain name DD_0 for IP_0 from a DNS database;

e) testing if the real domain name DD_0 is "no name";

f) testing if IP_0 is in an open relay database;

g) testing if IP_0 is in the allow_address database;

h) testing if IP_0 is in the prevent_address database,

i) requesting a connection with MTA_1;

j) waiting for a 220 reply from MTA_1 to acknowledge the requested connection;

k) waiting for a reply from either MTA_0 or MTA_1;

l) jumping to step o) if the reply is not from MTA_1;

- m) relaying the reply from MTA_1 to MTA_0;
- n) jumping to step k) to wait for a new reply;
- o) jumping to step u) if the reply from MTA_0 is not a HELO;
- p) extracting the declared domain name D_0 from the reply;
- q) testing if the declared domain name D_0 is the same as D_1;
- r) testing if the declared domain name D_0 of MTA_0 does not match the real domain name DD_0 of MTA_0 AND the declared domain name D_0 of MTA_0 is in the suspect_domain database;
- s) relaying the HELO reply from MTA_0 to MTA_1;
- t) jumping to step k) to wait for a new reply;
- u) jumping to step aa) if reply from MTA_0 is not a MAIL;
- v) extracting the from-address A_0;
- w) testing if A_0 is in the bad_from database;
- x) testing if DD_0 does not match the domain of A_0 and the domain of A_0 is in the suspect_domain database;
- y) relaying MAIL reply to MTA_1;
- z) jumping to step k) to wait for a new reply;
- aa) jumping to step qq) if the reply from MTA_0 is not a RCPT;
- bb) extracting the to-address A_1;
- cc) testing if A_1 is in the no_filter database;
- dd) testing if A_0 matches A_1;
- ee) jumping to step nn) if t_allow OR t_no_filter OR NOT (t_prevent OR t_open OR t_DD_0 OR t_bad_from OR t_suspect_domain OR t_match);
- ff) logging time, A_0, A_1, and reason for rejection in the rejected_connection database;
- gg) jumping to step ll) if A_1 is not in the save_spam database;
- hh) looking up A'_1 in the diversion database;
- ii) substituting A'_1 for A_1 in the RCPT reply;
- jj) sending the modified RCPT reply to MTA_1;
- kk) jumping to step k) to wait for a new reply;
- ll) rejecting the connection with MTA_0 connection by sending a 550 reply to MTA_0;
- mm) jumping to step k) to wait for a new reply;

- nn) logging time and A_1 in the allowed_connection database;
 - oo) relaying RCPT reply from MTA_0 to MTA_1;
 - pp) jumping to step k) to wait for a new reply;
 - qq) jumping to step bbb) if the reply from MTA_0 is not DATA;
 - rr) relaying DATA reply to MTA_1;
 - ss) waiting for a 354 reply from MTA_1;
 - tt) relaying the 354 reply from MTA_1 to MTA_0;
 - uu) waiting for the data from MTA_0;
 - vv) relaying the data from MTA_0 to MTA_1;
 - ww) waiting for a .r\n from MTA_0;
 - xx) relaying the .r\n from MTA_0 to MTA_1;
 - yy) waiting for a 250 reply from MTA_1;
 - zz) relaying the 250 reply to MTA_0;
 - aaa) jumping to step k) to wait for a new reply;
 - bbb) jumping to step eee) if reply from MTA_0 is not RSET, SEND, SCML, SAML, VRFY, NOOP, EXPN, HELP, or TURN;
 - ccc) relaying the reply to MTA_1;
 - ddd) jumping to step e) to wait for a new reply;
 - eee) jumping to step jjj) if the reply from MTA_0 is not a QUIT;
 - fff) relaying the QUIT reply to MTA_1;
 - ggg) waiting for a 221 reply from MTA_1;
 - hhh) relaying the 221 reply from MTA_1 to MTA_0;
 - iii) jumping to step a) to wait for a new connection;
 - jjj) sending a 500 reply to MTA_0 to signal a syntax error; and
 - kkk) jumping to step a) to wait for a new connection.
16. (Cancelled)
17. (Cancelled)